(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 January 2003 (03.01.2003)

PCT

(10) International Publication Number
WO 03/001504 A1

(51) International Patent Classification[7]: G10L 15/00, G06F 3/16, 17/20, 17/27, 17/28

(21) International Application Number: PCT/AU02/00802

(22) International Filing Date: 19 June 2002 (19.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PR5851 21 June 2001 (21.06.2001) AU

(71) Applicant *(for all designated States except US)*: **KAZ GROUP LIMITED** [AU/AU]; Level 7, 66 Wentworth Avenue, Sydney, NSW 2010 (AU).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ESTIVAL, Dominique** [FR/AU]; Level 7, 66 Wentworth Avenue, Sydney, NSW 2010 (AU). **HUTCHISON, Ben** [AU/AU]; Level 7, 66 Wentworth Avenue, Sydney, NSW 2010 (AU).

(74) Agent: **WALLINGTON-DUMMER**; P.O. Box 297, Rydalmere, NSW 1701 (AU).

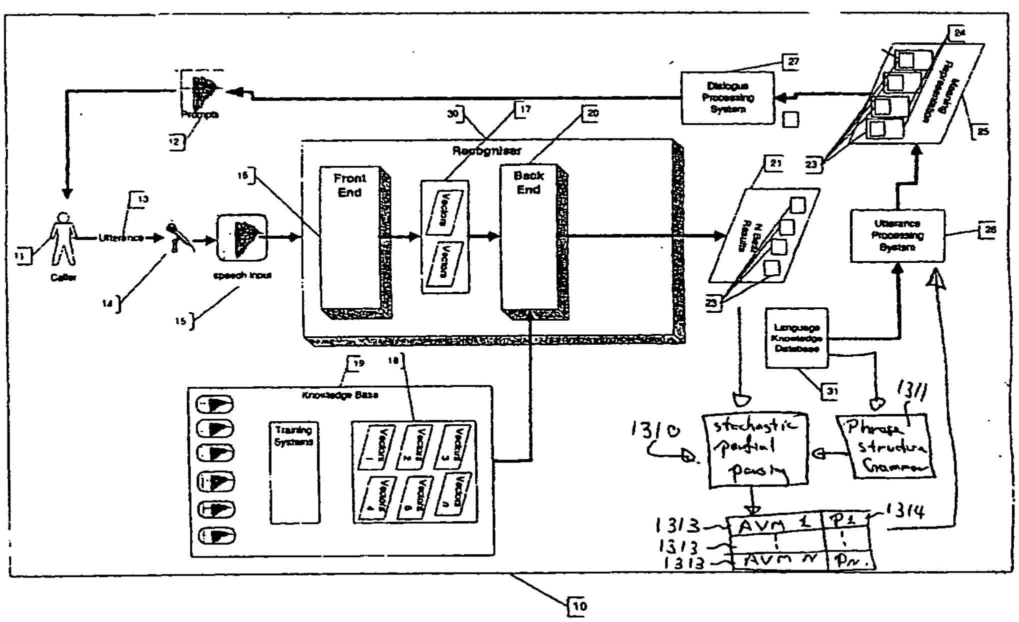
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: STOCHASTIC CHUNK PARSER



(57) Abstract: In a speech recognition system of the type adapted to process utterances from a caller or user by way of components including a recogniser, an utterance processing system and a dialogue processing system so as to produce responses to said utterances, a method of parsing a data structure derived by a component of said system; said method comprising performing only a partial parse of said data structure.

WO 03/001504 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

STOCHASTIC CHUNK PARSER

The present invention relates to a stochastic chunk parser and, more particularly, to such a parser suited for use within an automated speech recognition system.

BACKGROUND

Automated speech recognition is a complex task in itself. Automated speech understanding sufficient to provide automated dialogue with a user adds a further layer of complexity.

In this specification the term "automated speech recognition system" will refer to automated or substantially automated systems which perform automated speech recognition and also attempt automated speech understanding, at least to predetermined levels sufficient to provide a capability for at least limited automated dialogue with a user.

A generalized diagram of a commercial grade automated speech recognition system as can be used for example in call centres and the like is illustrated in Fig. 1.

With advances in digital computers and a significant lowering in cost per unit of computing capacity there have been a number of attempts in the commercial marketplace to install such automated speech recognition systems implemented substantially by means of digital computers.

However, to date, there remain problems in achieving 100% recognition and/or 100% understanding in real time.

In particular parsing techniques are used to help take utterances input to the system so as to decide the likelihood of different interpretations being placed on those utterances.

Parsing techniques that produce complete parse trees and attribute value matrices have been used. Also, parsing techniques have been used which produce complete parse trees but may have skipped over one or more words. The grammars used in these solutions have been difficult to develop and maintain.

These problem has not been satisfactorily solved in the past due to the complicated nature of spoken language, containing, for example, repairs, pauses, discourse markers, asides, editing terms, etc. In addition, many words in spoken language are ignored by the application but are required to be included in grammars used by parsers searching for complete parsers.

It is an object of the present invention to address or ameliorate one or more of the abovementioned disadvantages.

BRIEF DESCRIPTION OF INVENTION

Accordingly, in one broad form of the invention there is provided in a speech recognition system of the type adapted to process utterances from a caller or user by way

of components including a recogniser, an utterance processing system and a dialogue processing system so as to produce responses to said utterances, a method of parsing a data structure derived by a component of said system; said method comprising performing only a partial parse of said data structure.

In a further broad form of the invention there is provided in a speech recognition system of the type adapted to process utterances from a caller or user by way of components including a recogniser, an utterance processing system and a dialogue processing system so as to produce responses to said utterances, a stochastic chunk parser for attribute grammars which searches for meaningful words and phrases within said utterances and from these builds attribute value matrices.

Preferably said parser utilizes structure and location of said phrases to estimate the probability of the utterance in which they are found.

Preferably said parser receives as input an N-best list derived from said recogniser.

Preferably said attribute value matrices attach respective probabilities to respective attribute value matrices by combining phrase structure probabilities with class-based N-gram language modeling.

Preferably said method is applied to a data structure.

Preferably said data structure comprises a N-best list.

Preferably said data structure comprises a word lattice.

BRIEF DESCRIPTION OF DRAWINGS

Embodiments of the present invention will now be described with reference to the accompanying drawings wherein:

Fig. 1 is a generalized block diagram of a prior art automated speech recognition system;

Fig. 2 is a generalized block diagram of an automated speech recognition system suited for use in conjunction with an embodiment of the present invention;

Fig. 3 is a more detailed block diagram of the utterance processing and dialogue processing portions of the system of Fig. 2;

Fig. 4 is a more detailed block diagram of the system of Fig. 2 incorporating a stochastic chunk parser in accordance with a first preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

With reference to Fig. 2 there is illustrated a generalized block diagram of an automated speech recognition system 10 adapted to receive human speech

derived from user 11, and to process that speech with a view to recognizing and understanding the speech to a sufficient level of accuracy that a response 12 can be returned to user 11 by system 10. In the context of systems to which embodiments of the present invention are applicable the response 12 can take the form of an auditory communication, a written or visual communication or any other form of communication intelligible to user 11 or a combination thereof.

In all cases input from user 11 is in the form of a plurality of utterances 13 which are received by transducer 14 (for example a microphone) and converted into an electronic representation 15 of the utterances 13. In one exemplary form the electronic representation 15 comprises a digital representation of the utterances 13 in .WAV format. Each electronic representation 15 represents an entire utterance 13. The electronic representations 15 are processed through front end processor 16 to produce a stream of vectors 17, one vector for example for each 10ms segment of utterance 13. The vectors 17 are matched against knowledge base vectors 18 derived from knowledge base 19 by back end processor 20 so as to produce ranked results 1-N in the form of N best results 21. The results can comprise for example subwords, words or phrases but will depend on the application. N can vary from 1 to very high values, again depending on the application.

An utterance processing system 26 receives the N best results 21 and begins the task of assembling the results into a meaning representation 25 for example based on the data contained in language knowledge database 31.

The utterance processing system 26 orders the resulting tokens or words 23 contained in N-best results 21 into a meaning representation 25 of token or word candidates which are passed to the dialogue processing system 27 where sufficient understanding is attained so as to permit functional utilization of speech input 15 from user 11 for the task to be performed by the automated speech recognition system 10. In this case the functionality includes attaining of sufficient understanding to permit at least a limited dialogue to be entered into with user/caller 11 by means of response 12 in the form of prompts so as to elicit further speech input from the user 11. In the alternative or in addition, the functionality for example can include a sufficient understanding to permit interaction with extended databases for data identification.

Fig. 3 illustrates further detail of the system of Fig. 2 including listing of further functional components which make up the utterance processing system 26 and the dialogue processing system 27 and their interaction. Like components are numbered as for the arrangement of Fig. 2.

The utterance processing system 26 and the dialogue processing system 27 together form a natural language processing system. The utterance processing system 26 is event-driven and processes each of the utterances 13 of caller/user 11 individually. The dialogue processing system 27 puts any given utterance 13 of caller/user 11 into the context of the current conversation (usually in the context of a telephone conversation). Broadly, in a telephone answering context, it will try to resolve the query from the caller and decide on an appropriate answer to be provided by way of response 12.

The utterance processing system 26 takes as input the output of the acoustic or speech recogniser 30 and produces a meaning representation 25 for passing to dialogue processing system 27.

In a typical, but not limiting form, the meaning representation 25 can take the form of value pairs. For example, the utterance "I want to go from Melbourne to Sydney on Wednesday" may be presented to the dialogue processing system 27 in the form of three value pairs, comprising:

1. Start; Melbourne
2. Destination; Sydney
3. Date; Wednesday

where, in this instance, the components Melbourne, Sydney, Wednesday of the value pairs 24 comprise tokens or words 23.

With particular reference to Fig. 3 the recogniser 30 provides as output N best results 21 usually in the form of tokens or words 23 to the utterance processing system 26 where it is first disambiguated by language model 32. In one form the language model 32 is based on trigrams with cut off.

Analyser 33 specifies how words derived from language model 32 can be grouped together to form meaningful phrases which are used to interpret utterance 13. In one form the analyzer is based on a series of simple finite state automata which produce robust parses of phrasal chunks - for example noun phrases for entities and concepts and WH-phrases for questions, dates. Analyser 33 is driven by grammars such as meta-grammar 34. The grammars themselves must be tailored for each application and can be thought of as data created for a specific customer.

The resolver 35 then uses semantic information associated with the words of the phrases recognized as relevant by the analyzer 33 to refine the meaning representation 25 into its final form for passing through the dialogue flow controller 36 within dialogue processing system 27.

The dialogue processing system 27, in this instance with reference to Fig. 3, receives meaning representation 25 from resolver 35 and processes the dialogue according to the appropriate dialogue models. Again, dialogue models will be specific to different applications but some may be reusable. For example a protocol model may handle greetings, closures, interruptions, errors and the like across a number of different applications.

The dialogue flow controller 36 uses the dialogue history to keep track of the interactions.

The logic engine 37, in this instance, creates SQL queries based on the meaning representation 25. Again it will be dependent on the specific application and its domain knowledge base.

The generator 38 produces responses 12 (for example speech out). In the simplest form the generator 38 can utilize generic text to speech (TTS) systems to produce a voiced response.

Language knowledge database 31 comprises, in the instance of Fig. 3, a lexicon 39 operating in conjunction with database 40. The lexicon 39 and database 40 operating in conjunction with knowledge base mapping tools 41 and, as appropriate, language model 32 and grammars 34 constitutes a language knowledge database 31 or knowledge base which deals with domain specific data. The structure and grouping of data is modeled in the knowledge base 31.

Database 40 comprises raw data provided by a customer. In one instance this data may comprise names, addresses, places, dates and is usually organised in a way that logically relates to the way it will be used. The database 40 may remain unchanged or it may be updated throughout the lifetime of an application. Functional implementation can be by way of database servers such as MySQL, Oracle, Postgres.

As will be observed particularly with reference to Fig. 3, interaction between a number of components in the system can be quite complex with lexicon 39, in particular, being used by and interacting with multiple components of System 10.

With reference to Fig. 4 a stochastic chunk parser 1310 in accordance with a first preferred embodiment of the invention is illustrated in block diagram form. Parser 1310 takes as its input the output of recogniser 30 and, with the assistance of a phrase structure grammar 1311 it produces a set 1312 of attribute value matrices, each having associated with it a probability 1314. In this instance the set 1312 comprises N attribute value matrices 1313 together with a corresponding respective N probabilities 1314. In one particular form the probabilities are calculated by combining phrase structure probabilities with class-based N-gram language modeling,

thereby to allow the calculation of probabilities of partial parses.

The Stochastic Chunk Parser for Attribute Grammars searches for meaningful words and phrases, and from these builds attribute value matrices (AVM). It also uses the structure and location of these phrases to estimate the probability of the utterances they are found in.

The solution differs from previous attempts in that no complete parse of the sentence is performed. This means the parser does not attempt to find structural analysis for repairs, discourse markers, asides, editing terms, and other words which do not contribute to the construction of AVMs.

The Stochastic Chunk Parser for Attribute Grammars 1310 takes as its input a scored N-best list 21 created by a speech recogniser 30. From this, the Parser uses partial parsing to create a set of AVMs. The parsing is stochastic, so that probabilities are associated with each AVM. This is used to calculate the most likely AVM from the speech input.

The Parser 1310 uses phrase structure grammars 1311, and proceeds from the bottom up. As each phrase structure rule is matched AVMs 1313are created representing the meaning of the phrase. At the end of structural analysis there is a sequence of terminal and non-terminal categories; the former represent unparsed words, the latter

parsed structures. The probabilities of these sequences are calculated in a novel manner, combining phrase structure probabilities with class-based n-gram language modelling. The internal structure of phrases are calculated using probabilistic parsing techniques, and the product of these is multiplied with the probability of the sequence of categories, which is estimated using an N-gram language model.

The same principle can be used to do parsing on data structures other than N-best lists, for example word lattices.

The advantage of the solution is that allows the robust stochastic parsing of spoken language. The grammars required by the Parser are considerably simpler than those used by parsers doing complete analyses. This means that application development time is significantly reduced.

The principle underlying the solution is that of using a mixture of phrase probabilities and N-gram language modelling to allow the calculation of probabilities of partial parses.

Variations that embody the concept include other methods of combining phrase structure probabilities with the probability of a series of tokens, including fixed and variable length language models, word based language models that backoff to class based, etc.

Annexure A provides additional background on the concept of probabilistic partial parsing and is incorporated herein by cross reference and so as to form part of this specification.

The above describes only some embodiments of the present invention and modifications, obvious to those skilled in the art, can be made thereto without departing from the scope and spirit of the present invention.

Annexure A

1/5

Probabilistic Partial Parsing

Introduction

Probabilistic Parsing assigns probabilities to parse trees. This has two possible functions in the field of spoken language understanding:

1. Re-scoring speech recognition candidates.
2. Choosing between ambiguous parses.

The first function addresses what has been a concern for some time: how can linguistic knowledge be used to improve recognition performance? Language modelling using N-grams is one way to incorporate statistics on language use, but it is unable to sensibly cope with syntactic relations other than the most local.

To date, 2. has not been an issue for use since our parser has been deterministic. Recall that a deterministic parser producesd one and only one parse for each sentence, while a non-deterministic parser is capable of producing multiple parses for a sentence. Note that since we are doing partial parsing we are always guaranteed of finding a partial parse for a sentence, however "partial" may be "zero", in which case no parsing whatsoever may have been done to produce the partial parse. While deterministic parsers are the fastest, and to date have been sufficient for our purposes, it should be kept in mind that non-deterministic parsing approaches may be required in the future

Non-deterministic parsing

In this section I give an example where non-deterministic parsing is required to produce the desired analysis. (Note that the example is meant to be illustrative only – the problem could also be solved by including "turn_off" in the lexicon.) Consider the following two verb phrases.

(1) turn off the light
(2) turn off the road

The correct chunking off these would be[1]:

(3) (turn off) (the light)
(4) (turn) (off (the road))

Consider the following analysis grammar:

Level 1

VERB → VB
    then Action = turn
    end.

PHRASALVERB → VB IN
    if 1.v eq turn and 2.v eq off
    then Action = extinguish
    end.

NP → [DT] NN
    then Object = 2.v
    end.

[1] According to the analysis in Rodney Huddlestone's *Introduction to the Grammar of English*.

14 – A2

2/5

Level 2

PP → IN NP
if 1.v eq off and 2.Object eq road
then DirectionOff = 2.Object
end.

The above grammar will produce the parses:

(5) * (turn) off (the light)
(6) (turn) (off (the road))

If the precedence of the first two rules is swapped, the grammar will instead produce

(7) (turn off) (the light)
(8) * (turn off) (the road)

No re-ordering of the rules will produce the correct chunking of both verb phrases. This is because the correct Level 1 chunking of *turn* is co-dependent with the correct chunking of the complement phrase. One solution to this would be to introduce context-sensitivity into the grammar, but this is undesirable. There are strong arguments that it makes sense to keep both *(turn) off (the road)* and *(turn off) (the road)* alive as candidate Level 1 chunkings until the applicability of Level 2 rules has been checked.

To summarise:

- Probabilistic parsing allows choosing amongst competing candidate parses;
- Producing competing candidate parses requires non-deterministic parsing.

I have tried to provide a brief sketch of how non-deterministic parsing may be required when we attempt more complicated semantic analyses. I will now discuss interesting issues related to non-deterministic partial parsing.

Non-deterministic parsing and lattices

Non-determinism parsing has an interesting connection with lattices that is worth discussing. In order to keep future possibilites open, non-deterministic parsing essentially treats all grammar rules as optional: a rule may be applied, or it may be skipped in order to allow future rules the possibility of applying. Since this leads to a binary branching of the search space, it may be supposed that this leads to an exponential explosion of the search space. This belief is misguided however, since the interdependence of rules, both at the same level, and across levels, places a large restriction on the number of parses. Furthermore, storing possible hypotheses in a chart allows efficient searching for possible parses. For example, considering the grammar above as non-deterministic, the following parse for *turn off the road* would be constructed.

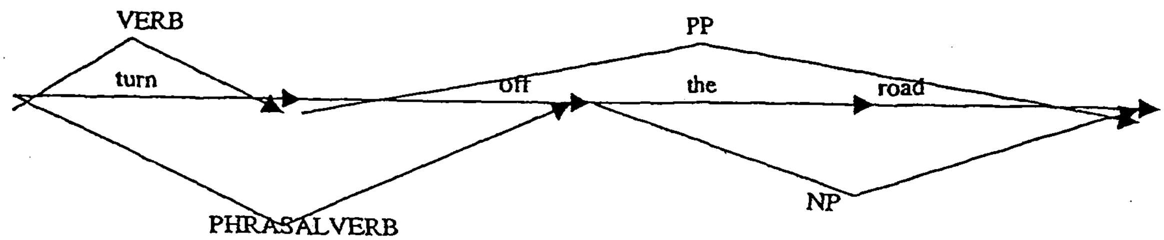
	PP: *off the road*		
PHRASALVERB: *turn off*			
VERB: *turn*		NP: *the road*	
VB	IN	DT	NN
turn	off	the	road

When all possible parses have been stored in a chart a Viterbi-style search algorithm can locate the most probable partial parse. In the above example, this would hopefully show that the parse *(turn) (off (the road))* is more probable than any of *(turn off) (the road)*, *(turn) off (the road)*, *(turn) off the road*, *turn off (the road)*, *turn ((off) the road)* or the "zero" parse *turn off the road*. This raises an interesting situation: since a chart can be thought of as a lattice, and so non-deterministic parsing can be seen as taking an

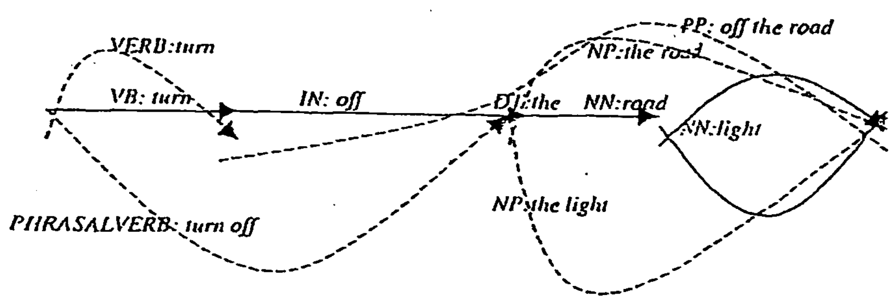
14 - A3

3/5

utterance represented as a simple lattice where each arc represents a single word and adding arcs corresponding to phrasal categories. This creates a lattice of all partial parses.



Furthermore, the input to this need not be a single utterance, but can be a lattice of recognition hypotheses. For example, the lattice below shows the results of non-deterministically parsing a recognition lattice which initially contained just the two traversals: *turn off the road* and *turn off the light* (intoduced arcs are shown in dotted line).



Once this lattice has been constructed, the Viterbi search should produce one of *(turn off) (the light)* and *(turn) (off (the road))* as the most likely parse.

To summarise:

- nondeterministic parsing is done most efficiently using chart parsing;
- chart parsing is equivalent to adding arcs to a lattice;
- hence parsing of recognition hypothesis lattices is a natural extension to parsing single utterances.

Non-deterministic partial parsing and maximal parses

The above discussion holds for both full and partial parsing. In both cases the chart/lattice containing all parsed structures is constructed. The only difference is that a full parse requires the existence of an arc leading straight from the START node of the lattice to the end NODE. Probabilistic full parsing requires selecting the most probable of all such arcs. Probabilistic partial parsing requires a Viterbi search through the lattice to find the most probable path from the START node to the END node. Depending on the modelling techniques and the available statistics, it is theoretically possible that the Viterbi search may produce a partial parse which is just a "zero" parse, i.e. just a sequence of words when in fact a phrasal arc was present in the parse lattice. One way to avoid this is to only allow partial parses which are maximal. By maximal, I mean that there exists no other parse which is identical except for the application of another rule. Perhaps the easiest way to preclude the Viterbi search from returning non-maximal parses is to cleverly prune the lattice of arcs in such a way that all traversals of the lattice produce maximal parses. Care must be taken when pruning in order not to exclude any maximal parses. Reflection led me to the following pruning procedure for achieving this. In fact, it may be that the property of being included in a

14- A4

4/5

maximal parse is not sufficiently local for an elegant pruning algorithm to exist. So in the meantime we must trust that the training data will never prefer a non-maximal parse.

Probabilistic partial parsing – the nitty gritty

Unable to find any established results in the literature on probabilistic partial parsing, in this section I derive the formulas required from scratch. Recall that the Inside and Outside probabilities are used in probabilistic parsing. The Inside probability of a subtree is the probability of all the expansions within that subtree, whereas the Outside probability is the probability of that subtree ocurring within the context of the sentence. In the case of complete parses, the supertrees' phrase expansion probabilities are used to calculate the Outside probabilities. However since we are guaranteed only partial parses, we often will not have a supertree. In these cases we shall use category-based N-gram language modelling to calculate the Outside probabilities, where the categories can be either phrasal or lexical.

So, assume that a partial parse produces a sequence of categories C_i. The probability of the parse is calculated by:

$$\boxed{P(C_{1..n}) = \prod_{i=1}^{n} P_{In}(C_i) P_{Out}(C_i)}$$

Where P_{Out} is calculated using (backoff) trigram probabilities, and P_{In} is the inside probability of the category, which in the case of lexical categories defaults to the probability of the word given the word class. Note that under this model probabilistic parsing replaces language modelling. However it would also be possible to do some language modelling earlier on in order to reduce the candidate space, then ignore the language modelling results and do probabilistic parsing.

In practise, the outside probabilities can be calculated using the Acoustic group's language model class. This means we must just calculate the inside probabilities, which is quite a simple taskof multiplying probabilities of phrase expansions and lexical instantiations. Note also that it will also be easy to do speech repair by doing lattice pruning, and also ALARM processing is a simple case of lattice pruning. This shows we are ready to accept lattices from the acoustics group. Even more, if we decide to proceed with a non-deterministic partial parser then lattice parsing is not just possible but desirable!

Data, damned data!

In order to calculate the inside probabilities of phrase trees, and also the outside probabilities using N-gram probabilities of phrasal categories, a corpus of parsed sentences must be available. I propose that we develop a semi-automatic procedure whereby the user is presented with all *maximal* parses of each utterance in the corpus. They then make choose the correct one, and this information results in frequency statistics for N-grams and phrase structure expansions.

A non-statistical approach to disambiguation

Since we will not always have parsed corpora, the analyser should be able to disambiguate using information from the Metagrammar. Here I propose a non-deterministic method of doing analysis for speech recognition applications. Crucial to the proposal is that the grammar for analysis should be derived from a speech recognition grammar in which certain chunks, or phrases, have been indicated by the grammar to be meaningful.

Advantages:

- Automatically and efficiently disambiguates problem phrases
- Eliminates the need for conditions in the grammar
- Eliminates the dependence on rule order

Before describing the algorithm, I first introduce the concept of [illegible] *chunks*.

(4 – A5

5/5

- By *top chunk* I mean a rule of the speech recognition grammar that has semantic tags, but all of the rules that reference do not have semantic tags.

During analyser compilation:

1) Find each top chunk of the grammar.

During analysis:

2) Use chunk parsing as we currently do, except multiple hypotheses can be kept alive at each level.
3) After parsing, discard all hypotheses that are not top-level rules. These are cases that have already been disambiguated from context.
4) If ambiguities remain (i.e. if there exist overlapping top chunks), resolve them using the recognition grammar.

CLAIMS

1. In a speech recognition system of the type adapted to process utterances from a caller or user by way of components including a recogniser, an utterance processing system and a dialogue processing system so as to produce responses to said utterances, a method of parsing a data structure derived by a component of said system; said method comprising performing only a partial parse of said data structure.

2. In a speech recognition system of the type adapted to process utterances from a caller or user by way of components including a recogniser, an utterance processing system and a dialogue processing system so as to produce responses to said utterances, a stochastic chunk parser for attribute grammars which searches for meaningful words and phrases within said utterances and from these builds attribute value matrices.

3. The parser of Claim 2 wherein said parser utilizes structure and location of said phrases to estimate the probability of the utterance in which they are found.

4. The parser of Claim 2 or Claim 3 which receives as input an N-best list derived from said recogniser.

5. The parser of any previous claim wherein said attribute value matrices attach respective probabilities to respective attribute value matrices by combining phrase structure probabilities with class-based N-gram language modeling.

6. The method of Claim 1 applied to a data structure.

7. The method of Claim 6 wherein said data structure comprises a N-best list.

8. The method of Claim 6 wherein said data structure comprises a word lattice.

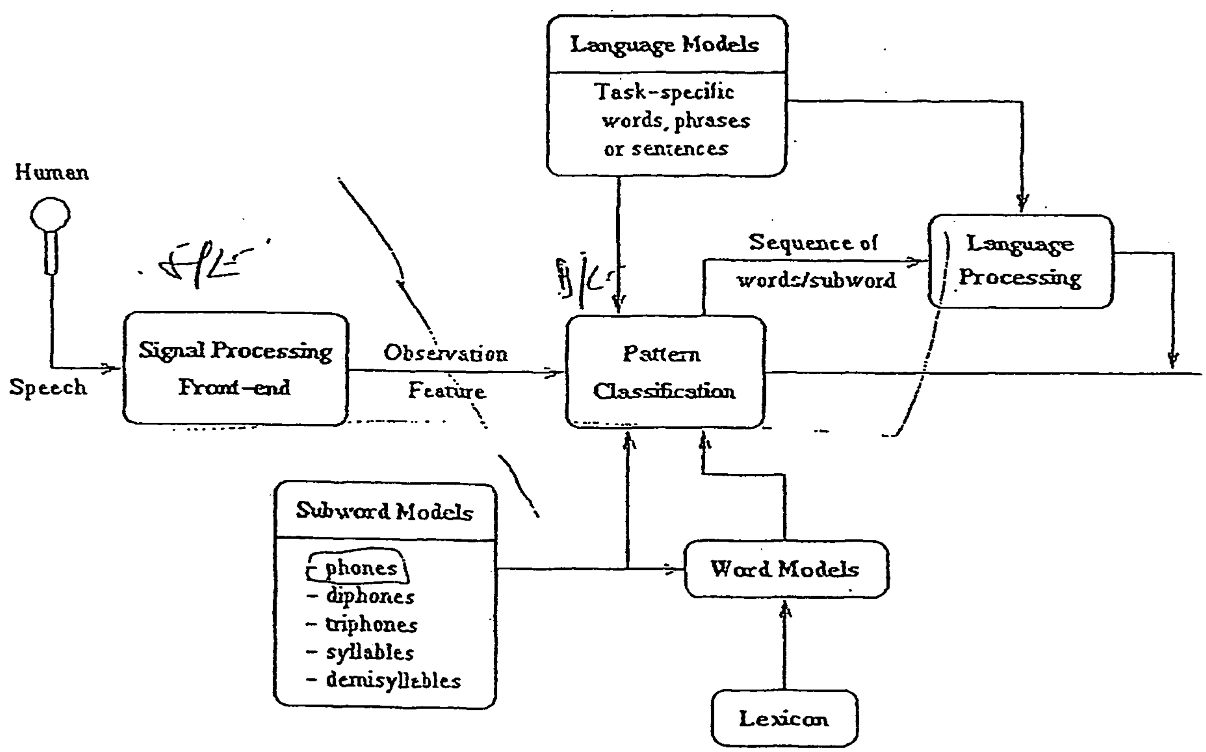

Language Models
Task-specific words, phrases or sentences
Human
Speech
Signal Processing Front-end
Observation Feature
Pattern Classification
Sequence of words/subword
Language Processing
Subword Models
phones
- diphones
- triphones
- syllables
- demisyllables
Word Models
Lexicon

Fig 1

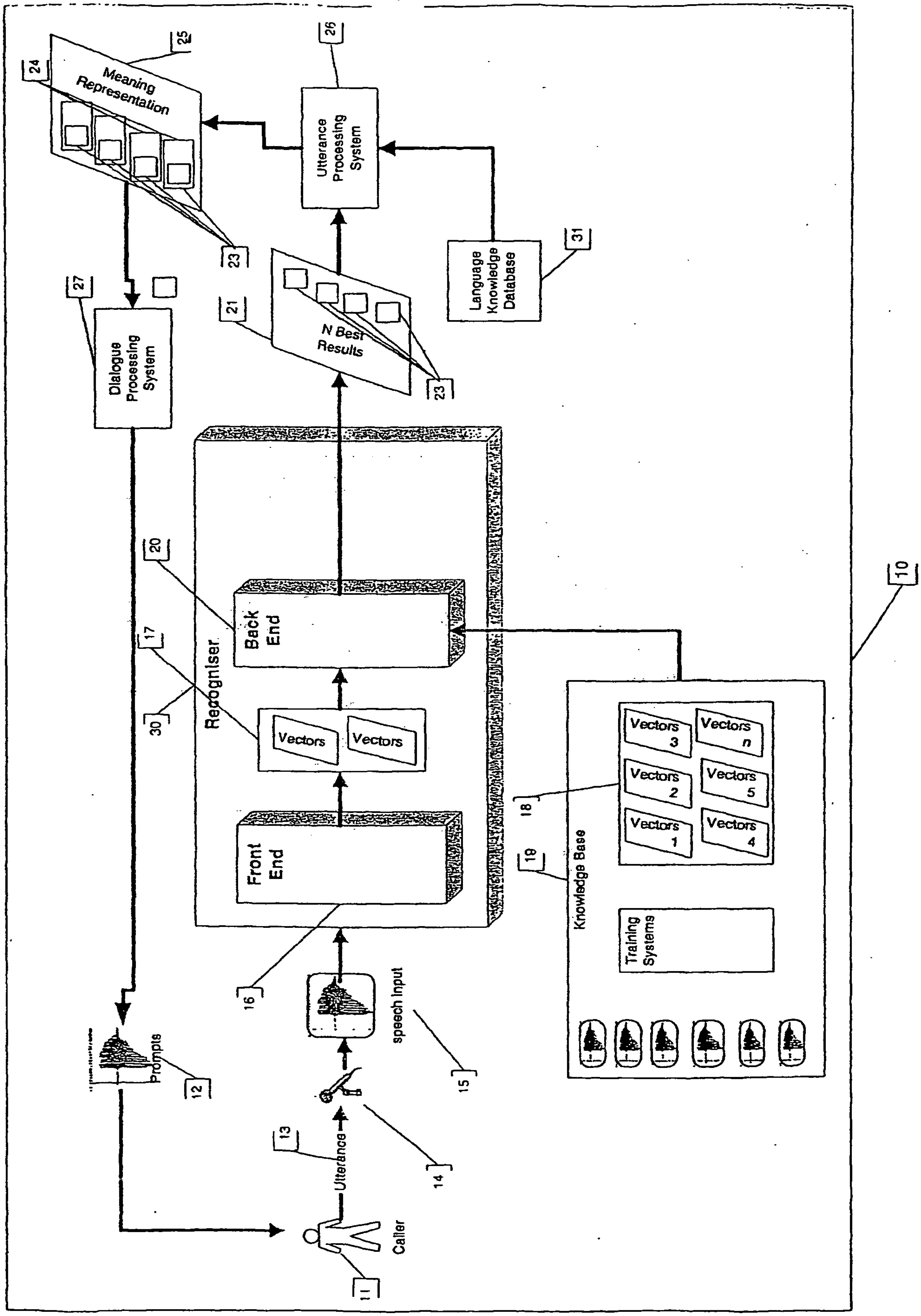
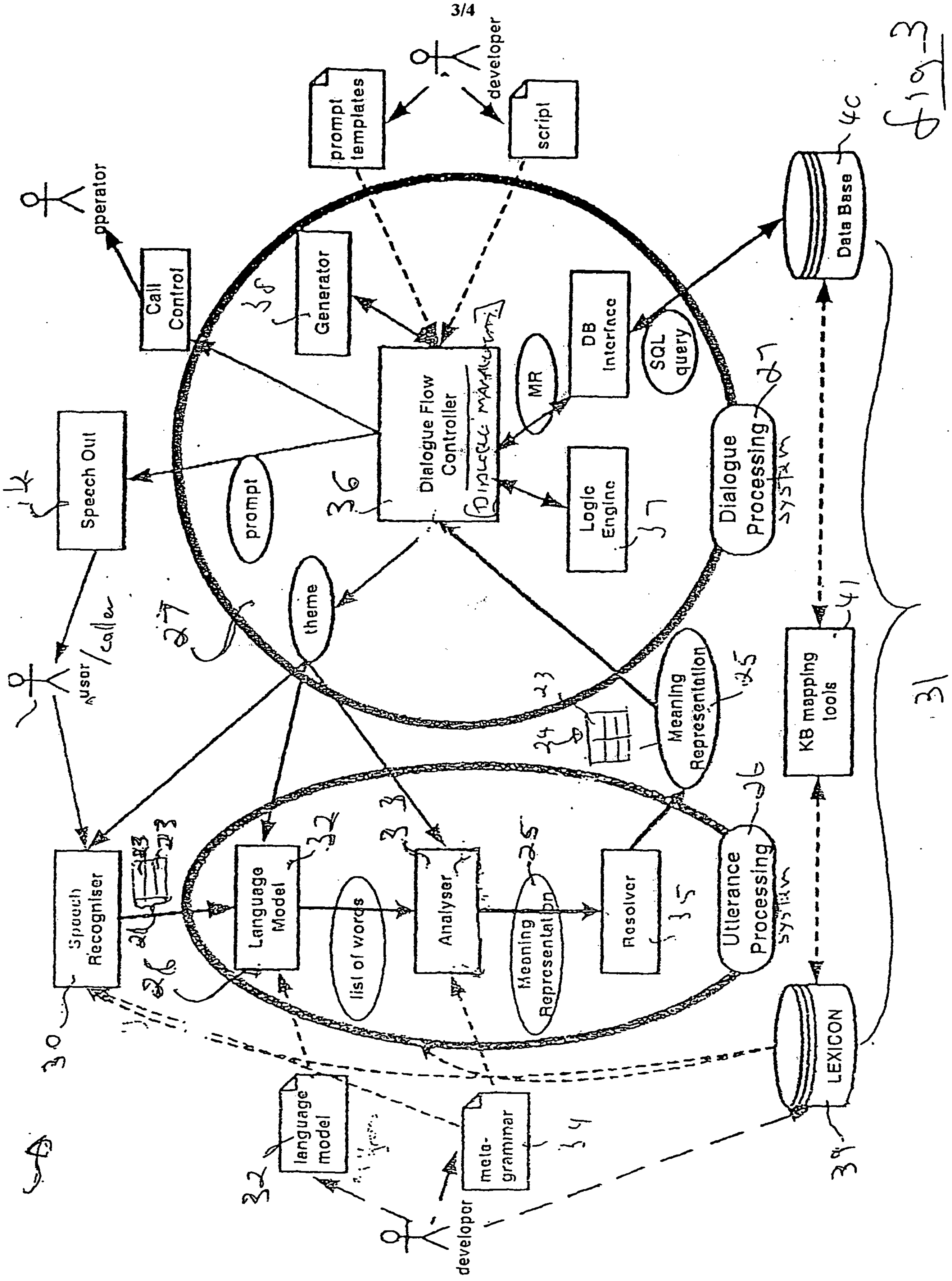

Meaning Representation
25
24
23
Utterance Processing System
26
Language Knowledge Database
31
Dialogue Processing System
27
N Best Results
21
23
Back End
20
Recogniser
17
30
Vectors
Vectors
Front End
16
Prompts
12
speech input
15
Utterance
13
14
Caller
11
Knowledge Base
19
18
Vectors 1
Vectors 2
Vectors 3
Vectors 4
Vectors 5
Vectors n
Training Systems
10

Figure 2


developer
prompt templates
script
operator
Call Control
Generator
Speech Out
Dialogue Flow Controller
MR
DB Interface
SQL query
Logic Engine
prompt
theme
Dialogue Processing
Data Base
user/caller
Meaning Representation
KB mapping tools
Speech Recogniser
Language Model
list of words
Analyser
Meaning Representation
Resolver
Utterance Processing
LEXICON
language model
meta-grammar
developer

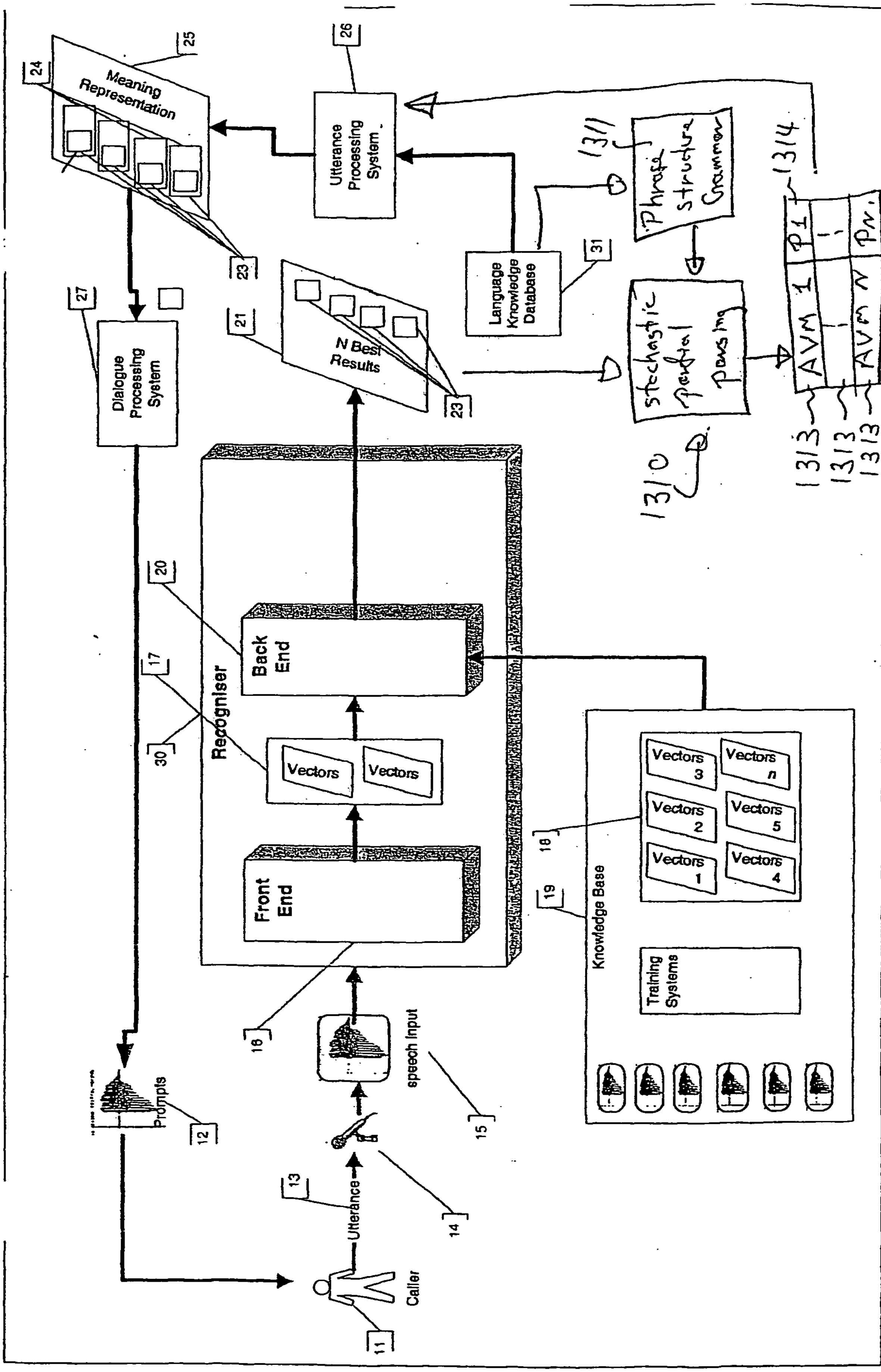

Meaning Representation
Utterance Processing System
Dialogue Processing System
N Best Results
Language Knowledge Database
Phrase Structure Grammar
stochastic partial parsing
AVM 1
AVM N
P1
Pn
Recogniser
Front End
Back End
Vectors
Vectors
Knowledge Base
Vectors 1
Vectors 2
Vectors 3
Vectors 4
Vectors 5
Vectors n
Training Systems
Prompts
Caller
Utterance
speech Input
10
11
12
13
14
15
16
17
18
19
20
21
23
23
24
25
26
27
30
31
1310
1311
1313
1313
1313
1314

Figure 4

INTERNATIONAL SEARCH REPORT	International application No. PCT/AU02/00802

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. ^{7}: G10L 15, G06F 3/16, G06F 17/20, G06F 17/27, G06F 17/28

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
DWPI IPC Marks and Keywords: Speech recognition, Voice recognition, Auto+, Adapt+, Adjust+, Learn+, Teach+, Feedback, Test+, Gaug+, Correlat+, Pars+, Dialog+, Conversat+, Score, Mark, Grade

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00/26901, A (DRAGON SYSTEMS INC.) 11 May 2000 Whole Document	1-8
X	WO 00/78022, A (TELSTRA R & D MANAGEMENT) 21 December 2000 Whole Document	1-8
P,X	WO 02/33583, A (TELSTRA R & D MANAGEMENT) 25 April 2002 Whole Document	1-8

[X] Further documents are listed in the continuation of Box C [X] See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 August 2002	5 SEP 2002 5 SEP 2002
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	Authorized officer J.W. THOMSON Telephone No : (02) 6283 2214

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU02/00802

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0977173, A(TEXAS INSTRUMENT INC.) 30 July 1999 Whole Document	1-8

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member			
WO	200026901	EP	1046156		
WO	200078022	AU	200050545	EP	1192789
WO	200233583	AU	20000824		
EP	977173	US	5988099		

END OF ANNEX